C-ALB - Collaborative Assembly Line Balancing: A New Approach in Cobot Solutions

Giovanni Boschetti
Faculty of Engineering- University of Padova
Maurizio Faccio (maurizio.faccio@unipd.it)
Faculty of Engineering- University of Padova
Mattia Milanese
Faculty of Engineering- University of Padova
Riccardo Minto
Faculty of Engineering- University of Padova

Research Article

Keywords: Collaborative robots, Assembly systems, Task allocation, DFCA

Posted Date: February 25th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-213225/v1

Version of Record: A version of this preprint was published at The International Journal of Advanced Manufacturing Technology on July 15th, 2021. See the published version at https://doi.org/10.1007/s00170-021-07565-7.

Noname manuscript No.
(will be inserted by the editor)

C-ALB - Collaborative Assembly Line Balancing: a new approach in cobot solutions

Giovanni Boschetti [1] · Maurizio Faccio [1]* · Mattia Milanese [2] · Riccardo Minto [1]

Received: date / Accepted: date

Abstract Collaborative robots can be a proper solution to improve the throughput of manual systems without reducing their flexibility. To effectively use cobots in productive systems it is fundamental to develop a suitable task allocation model that considers collaboration. Hence, we present a model for collaborative assembly line balancing (C-ALB) which considers paralleling tasks and collaboration in the balancing resolution. Indexes that take into account both the product and process characteristics are defined to evaluate the quality of the proposed task allocation model and comparing it to others. The results confirm the influence of the product characteristics on the system performance, leading to the definition of a new paradigm for product design.

Keywords Collaborative robots · Assembly systems · Task allocation · DFCA

1 Introduction

One of the most common challenges in modern production systems is the need to achieve a high level of flexibility without reducing the throughput. Indeed, the current market requires customized products in great quantity i.e., mass customization [1], and assembly is more affected than other technologies due to its position in the production process [2]. For this reason, the adoption of flexible systems capable to adapt to changes in products and volumes is fundamental [3].

To face this challenge, an immediate solution could be the adoption of manual assembly systems (MAS). Despite the advantages provided [4], the performance achievable by these systems is limited, with accuracy and repeatability needing further improvements. Moreover, the productivity of these systems is influenced by ergonomic problems [5].

On the other hand, automated systems [6] present higher throughput and better quality of the final product, reducing at the same time the cost of labor. Among the several improvements of flexible assembly systems (FAS) [7, 8], collaborative assembly systems (CAS) [9] are one of the solutions that could be implemented to achieve the requested flexibility. As stated by Takata et al. [10], collaborative robots, or cobots, may help to implement a dynamic productive cell, capable of sustaining a multi-model production, with the greater potential to adapt to model and volume changes.

To achieve the maximum performance from these systems, and therefore develop a cost-effective solution, it is important to correctly assign the tasks between the resources. Indeed, the idle times of a balanced assembly system are as low as possible, improving the collaboration between cobots and human operators. The activity of assigning the tasks to the stations/resources while optimizing some criterion and avoiding the violation of a certain number of constraints is called assembly line balancing (ALB) problem [11]. ALB problems could be grouped into two main categories: the first aims to mini-

Giovanni Boschetti
E-mail: giovanni.boschetti@unipd.it

Maurizio Faccio
E-mail: maurizio.faccio@unipd.it

Mattia Milanese
E-mail: mattia.milanese@unipd.it

Riccardo Minto
E-mail: riccardo.minto@phd.unipd.it

1
Department of Management and Engineering, University of Padova, Stradella S. Nicola, 3, Vicenza, Italy

2
Department of Industrial Engineering, University of Padova, Via Venezia, 1, Padova, Italy

mize the number of stations for a given cycle time, while the second aims to minimize the cycle time for a given number of stations. In the literature, the ALB problem is one of the most-studied topics for assembly systems, due to its practical implications, and many researchers presented effective solutions for exactly solving ALB methods [12].

However, traditional ALB models are not designed to take into account the possibility of two resources working in the same workstation. Despite previous ALB solutions have introduced a certain degree of parallelism [13] i.e., tasks distributed in parallel between the resources to improve the flexibility of the system, these methods aim to minimize the costs of the assembly line [14] since parallel stations require to double the equipment and fixtures. Furthermore, parallel stations are used to achieve cycle times lower than the task duration, which is not the aim of CAS. Indeed, cobots are deployed for various reasons [15] and the resources are not necessarily executing the same tasks in parallel, but usually their tasks are different.

Hence, this work aims to develop a mathematical ALB model designed for collaborative systems, hereafter called C-ALB, which considers the typical constraints of HRC:

– the task times may be different for each resource in the collaborative workcell;
– each resource may not be able to execute all tasks (*technological constraints*);
– precedence graph permitting, the resources can complete tasks in parallel.

The paper is organized as follows: section 2 presents the state of the art for ALB problems applied to collaborative systems and section 3 introduces the proposed model. Section 4 introduces the evaluation indexes used to estimate the quality of the obtained task allocation, whereas in section 5 the model is compared with a traditional ALBP model. The results are presented in section 6 and a practical application of the model is presented in section 7. Lastly, section 8 concludes the work.

2 Literature review

As previously described, mass customization leads to the necessity for performing, and expensive, automatic systems, hence the high practical implication of ALB problems. Since its first formalization by Salveson [19], many contributes are now available and several authors proposed different variations on the problem formulation, typically focusing on one or more specific features of the analyzed assembly system [20].

However, when considering the ALB problem applied to collaborative systems, the literature is limited. Table 1 reports a schematic representation of contributions on ALB applied to collaborative systems, focusing on works from the last decade, given the recent spread of this technology. The table groups previous researches on the basis of the objective of the task allocation, the input data required, and the proposed method. Regarding the latter, the majority of the considered works proposes a resolution by the development of frameworks for planning the collaborative process; despite this approach provides for easy-to-implement solutions, this advantage is counterbalanced by the qualitative nature of the framework. Even though the common objective of this works could be considered as improving the throughput of the system, our review identified further objectives, categorized in four main groups. In particular, Michalos et al. [21] proposed a framework with a multi-criteria approach, aiming to solve the problem of task assignment considering also the need to design an optimal layout for human-robot collaboration, and comparing the different process planning solutions in terms of productivity, ergonomics, process quality, and layout efficiency.

The interaction between human operators and cobots has always raised safety concerns, therefore it is reasonable that a high number of papers aims to improve operator safety during collaboration, while also considering the task ergonomics. These aspects are linked with the throughput in [27], where the authors proposed a planning method that regards both production time and physical strain. The authors introduced an objective function where a weight coefficient is defined, allowing the trade-off between these two parameters. This trade-off is also shown by Heydaryan et al. [35], who states that, despite a possible decrease in the throughput, collaboration can improve human ergonomics and reduce the risk of injury. Moreover, the framework proposed by Malik et al. [33] evaluates the complexity of assembly processes, also considering the assembly components characteristics e.g., geometrical and physical properties, and feeding mechanisms alongside the safety issues of the tasks. The framework aims to help the designer to define the most appropriate task assignment in a human-robot collaborative system.

On the other hand, the number of mathematical models adopted is minimum and the majority of them are based on heuristic algorithms. Bogner et al. [25] imply that heuristics allow to find a reasonable solution in a short amount of time. The authors propose a matheuristic algorithm to obtain a trade-off between the goodness of the solution and the computational time. A similar approach is adopted by Weckenborg

References	Authors	Year	Task Allocation				Input data				Resolution Method			Result
			Optimization of productivity and cycle time	Design workspace considering scheduling	Safety and ergonomics	Dynamic allocation and flexibility	Task data, graph, resources	CAD Data	Product Charateristic	Other/none	Framework	Genetic Algoritm	Heuristic	
[25]	Bogner et al.	2018	X				3 X						X	Developed a mathematical model for task allocation and resolved with Heuristic
[24]	Chen et al.	2013	X				X					X		Developed a mathematical model for offline and online task scheduling with restrict resources
[22]	Fechter et al.	2018		X			X	X					X	Developed a planning system that considers layout
[17]	Johannsmeier et al.	2017			X	X	X				X			Developed a planning system that results dynamic and flexible, also when unforeseen events happen
[21]	Michalos et al.	2018	X			X		X	X		X			Developed a planning system that considers layout
[26]	Nikolakis et al.	2018				X	X				X			Reduces time spent on a specific job
[27]	Pearce et al.	2018	X		X		X				X	X		Reduce makespan considering the physical strain
[23]	Ranz et al.	2017			X		X		X		X			Developed a method for task allocation based on resources capability
[10]	Takata et al.	2011	X			X				X	X			Developed a method to find the best task allocation in the manufacturing process with multiple scenarios
[18]	Tan et al.	2010	X				X				X			Shorter assembly time using collaboration
[30]	Tsarouchi et al.	2016	X				X				X			Developed a planning system that minimizes makespan and average resource utilization; introduced body gesture to ensure human and robot interaction
[28]	Weckenborg et al.	2019	X				X					X		The results show that cycle time decrease with the utilization of cobots, and consequently, there is a productivity gain (12%)
[29]	Gualtieri et al.	2019				X	X				X			Developed a framework for assembly system conversion and task allocation, evaluating different criteria

Table 1: Summary of the literature review on collaborative robots and ALB models in the last decade.

			Task Allocation				Input data				Resolution Method			
References	Authors	Year	Optimization of productivity and cycle time	Design workspace considering scheduling	Safety and ergonomics	Dynamic allocation and flexibility	Task data, graph, resources	CAD Data	Product Charateristic	Other/none	Framework	Genetic Algorihtm	Heuristic	Result
[34]	Tsarouchi et al.	2017		X	X		X		X		X			Developed a framework for task allocation that considers also layout
[31]	Weckenborg et al.	2019	X		X		X					X		Developed a mathematical model for task balancing that aims to minimize cycle time and improve ergonomics
[32]	Bruno et al.	2018			X		X				X			Task allocation exploiting different skill between robot and human, without considering work balancing
[33]	Malik et al.	2019			X		X		X		X			Task allocation based on task execution complexity
[35]	Heydaryan et al	2018			X		X		X		X			Proves that even if production time increase, collaboration improves human ergonomics and reduce risk of injury
[36]	Antonelli et al.	2017			X		X				X			Task allocation method to avoid resources overload
[37]	Dianatfar et al.	2019				X			X		X			With task allocation, it is possible to reduce human fatigue and workload without decreasing productivity
[38]	Yaphiar et al.	2019	X				X							Developed a mathematical model for task balancing in mixed-model assembly line

Table 1: (continue)

et al. [31], who deal with the problem of ergonomics and especially human fatigue issues by evaluating if the workload on the human operator is over a certain limit. However, despite presenting mathematical models based on integer linear programming (ILP) or mixed-integer linear programming (MILP), these works are focused on the comparison between the heuristic and mathematical approach, without presenting the influence of the product and process on the system performance, as seen in the considered input data, which in the majority of the studies is focused only on task data.

From the literature review it is possible to observe how ALB problem resolution is one of the most important challenges in human-robot collaboration systems design. As we have tried to highlight, several approaches have been proposed, with different aims and methods, though the majority of the previous works focus on frameworks as resolution approaches and considering only the process characteristics, without considering the product influence. The lack of quantitative models, which address both product characteristics such as the precedence graph and process characteristics such as the different task times between the resources, leads to the aim of this study to develop a mathematical model based on ILP with the objective to increase the throughput but also the shared time between the resources.

3 Task allocation model for CAS

The objective of this work is to define an optimal task allocation between the resources in a CAS context. In greater detail, the addressed problem is:

- to define a task allocation that minimizes the makespan ms, since the evaluation of the cycle time for a single station layout is meaningless;
- to identify the effects of the product and process characteristics on the task allocation in a collaborative scenario.

Therefore, this work aims to solve the linear optimization problem by minimizing the objective function ms.

To reach this goal, an optimization model based on linear programming is proposed considering the typical constraints of a cobot system intending to minimize the total makespan for the completion of all tasks, but also constraints related to the product and process characteristics of the system. The proposed C-ALB model is designed to optimize the collaboration between two resources, even if they are of the same type i.e., human-human or robot-robot assembly systems. However, this work will focus only on human-robot collaborative assembly systems.

3.1 Nomenclature

Input variables and parameters

J	Number of tasks
i, j	Task indexes $i, j = 1, ..., J$
K	Number of resources
k	Resource index $k = 1, ..., K$
P_j	Number of immediate and transitive predecessors of task j
p_j	Predecessor index $p_j = 1, ..., P_j$
S_j	Number of immediate and transitive successors of task j;
s_j	index $s_j = 1, ..., S_j$
d_j	Number of arcs for task j
$t_{j,k}$	Task time j for resource k [s]

Output variables

Optimization variables:

x_{jkt}	Assembly line balance decision variable [binary]
y_{ji}	Decision variable representing precedence between task j and task i [binary]

Objective function:

ms	Makespan [s]

Indexes

Indexes adopted to evaluate the evaluate the obtained task allocation:

$p_{\%}$	Parallelism [13] index
$t_{\%}$	Task time index
$m_{\%}$	Makespan index
$c_{\%}$	Collaboration index

Other variables used in this work

t	Temporal instant [s]
T	Temporal horizon
M^{BIG}	Big M [39]
T_{min}	Lower bound for the makespan [s]
T_{max}	Upper bound for the makespan [s]
U_k	Set of the unfeasible tasks for resource k
c	Cycle time [s]
T_{coll}	Collaboration time [s]

3.2 Hypotheses

The proposed model starts from the classic resolution of the ALB problem. Therefore, some of the hypotheses that characterize the model are retained in the proposed work, which are:

- mass production of one homogeneous product by performing J operations of a given product process (single-model line hypothesis);
- deterministic and integral operation time;
- each task is performed by only one resource.

As stated before, it is not optimal to apply the traditional model to a collaborative solution; therefore, other hypotheses should be defined to extend the resolution method. The hypotheses that characterize the model are:

- the number of resources K is equal to 2;
- the assembly line is composed by a single station consisting of one human operator and one cobot;
- collaborative resources that share workplace and task time.

3.3 C-ALB for Collaborative Assembly Systems

We now present our model which aims to minimize the makespan ms in collaborative assembly station.

$$min\, ms = \sum_{t=0}^{T}\sum_{k=1}^{K}(t + t_{jk})x_{jkt} \tag{1}$$

$$\text{Subject to: } \sum_{t=0}^{T}\sum_{k=1}^{K} x_{jkt} = 1 \quad \forall j \tag{2}$$

$$\sum_{t=0}^{T}\sum_{k=1}^{K}(t + t_{p_j k}) \cdot x_{p_j kt} \leq \sum_{t=0}^{T}\sum_{k=1}^{K} t \cdot x_{jkt} \quad \forall j, p_j \tag{3}$$

$$\sum_{j=1}^{J} x_{jkt} \leq 1 \quad \forall k, t \tag{4}$$

$$\sum_{t=0}^{T}\sum_{k=1}^{K}(t + t_{ik}) \cdot x_{ikt} - M \cdot y_{ji} \leq \sum_{t=0}^{T}\sum_{k=1}^{K} t \cdot x_{jkt} \quad \forall j, i, \quad j \neq i \tag{5}$$

$$\sum_{t=0}^{T}\sum_{k=1}^{K}(t + t_{jk}) \cdot x_{jkt} - M(1 - y_{ji}) \leq \sum_{t=0}^{T}\sum_{k=1}^{K} t \cdot x_{ikt} \quad \forall j, i, \quad j \neq i \tag{6}$$

$$x_{jkt} = 0 \quad \forall j \in U_k \tag{7}$$

$$x_{jkt}, y_{ji} \in \{0, 1\} \quad \forall j, k, t \tag{8}$$

where the optimization variable is represented by the binary variable x_{jkt}:

$$x_{jkt} = \begin{cases} 1 \text{ if task } j \text{ is performed by the resource } k \text{ at } t \\ 0 \text{ otherwise} \end{cases} \quad \forall j, k, t \tag{9}$$

and which by definition observes the *integrality* constraints in eq. (8). Similarly to previous ALB problem resolution methods, x_{jkt} is the task assignation variable; however, differently from previous formulations, our work introduces a third dimension (i.e. time t) to correctly evaluate ms (1). Moreover, the dimension k represents both the number of resource/station and the resource type, characterized by certain technological constraints and task times.

Eq. (2) represents the *occurence* constraints, which, alongside the *integrality* constraints, ensure that in each temporal instant each task is performed by exactly one resource k. As for all the other constraints, the introduction of the temporal dimension requires modifying the classic constraint as seen in classical resolutions such as the Patterson and Albracht method. Similarly, eq. (3) extends previous formulations of the *precedence* constraints. Indeed, given the definition of x_{jkt} in eq. (9) and the first set of constraints (2), the second member of (3) identifies the exact time instant when task j is performed by the resource k. Hence, the constraints ensure that task j starts after the execution of its predecessors i since its starting time should be greater than or equal to the corresponding ending of the execution of tasks i.

Eq. (4-6) are a further set of *precedence* constraints newly introduced in this work and defined with a focus on collaborative systems. Indeed, eq. (3) ensure that the precedence constraint is fulfilled only in the case of direct precedence between tasks i and j. Considering the precedence graph shown in Figure 1, for a given task j the other $J - 1$ tasks can be divided into the following groups:

- P_j direct and transitive predecessors of task j, red in Figure 1;
- S_j direct and transitive successors of task j, green in Figure 1;
- tasks which are not direct or transitive predecessors/successors of task j.

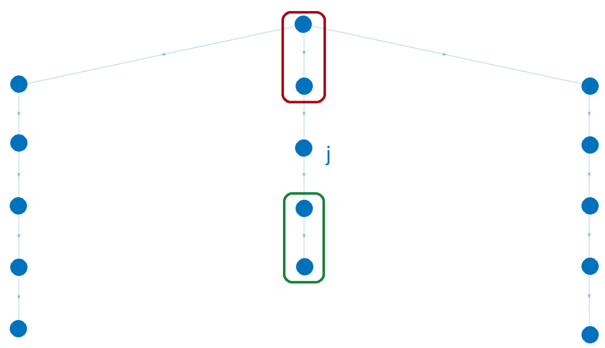


Fig. 1: Identification of the P_j predecessors and S_j successors for a task j considering a precedence graph.

Therefore, using only the constraints in eq. (2-3) do not consider the third group and tasks that belong to it could be performed at the same time by a resource k. To avoid this scenario, it is fundamental to ensure that the start time of task j is set after the end time of task i, that is:

$$start_j \geq end_i \quad \forall i > j \lor \forall i < j \tag{10}$$

where $start_j$ represent the start time of tasks j, and end_i represent the end time of task i and which corresponds to 2 sets of constraints, i.e. for j greater than i or j lesser than i. To correctly define these constraints, it is necessary to define the binary decision variable y_{ji} such as:

$$y_{ji} = \begin{cases} 1 \text{ if task } j \text{ is executed before task } i \\ 0 \text{ otherwise} \end{cases} \tag{11}$$

Hence, the constraints in (10) could be rewritten as:

$$start_j \geq end_i - M^{BIG} \cdot y_{ji} \tag{12}$$

$$start_i \geq end_j - M^{BIG} \cdot (1 - y_{ji}) \tag{13}$$

Hence, if task j is executed before task i, y_{ji} equals to 1 and eq. (12) is always true and eq. (13) ensures that task i starts after the complete execution of task j. On the contrary, if task j is executed after the end of task i, eq. (12) ensures that task j starts after the complete execution of task i, whereas eq. (13) is always true. Lastly, eq. (7) represents the technological constraints i.e. it limits the possible solutions by considering that some tasks j could not be carried out by the resource k. This is represented by the set U_k, whose elements u_{jk} are the tasks j that cannot be executed by resource k.

4 Product and process characteristics indexes

To better consider the influence of the product and process characteristics on the system performance for a considered task allocation, four indexes have been defined. These values take into account both input and output variables and both product and process characteristics and were divided as such:

- *parallelism index* $p_{\%}$: input parameter based on the precedence graph (*product* characteristic);
- *task time index* $t_{\%}$: input parameter based on the difference of the resources task times (*product and process* characteristic);
- *makespan index* $m_{\%}$: output parameter which evaluates the quality of the *ms* achievable with the considered task allocation (*process* characteristic);
- *collaboration parameter* $c_{\%}$: output parameter that represents the shared activity between the resources [9] (*process* characteristic).

It should be noted that these parameters do not define the model, thus they are not considered in the model description. This section aims to presents these four indexes, defining them and explaining briefly their influence on the system.

4.1 Parallelism index $p_{\%}$

The *parallelism index*, $p_{\%}$ is used to evaluate the number of parallel branches in a precedence graph, thus characterizing the product influence on the CAS. This is evaluated as the arithmetic mean of the ratio between the number of arcs for each task, hence:

$$p_{\%} = 1 - \frac{\sum_{j=1}^{J} \frac{d_j}{J-1}}{J} \in [0,1] \tag{14}$$

where d_j represents the number of arcs for task j and can be evaluated as:

$$d_j = P_j + S_j \tag{15}$$

and corresponds to the number of tasks that cannot be executed in parallel with task j. The proposed formulation allows to identify the parallelism degree for a precedence graph between two boundaries:

- if $p_{\%}$ is null, the tasks must be carried out in a sequential order as seen in Figure 2a; indeed:

$$\forall j, \quad d_j = n - 1, \quad p_{\%} = 0 \tag{16}$$

- if $p_{\%}$ is 1, the tasks can be carried out independently as seen in Figure 2b; thus:

$$\forall j, \quad d_j = 0, \quad p_{\%} = 1 \tag{17}$$

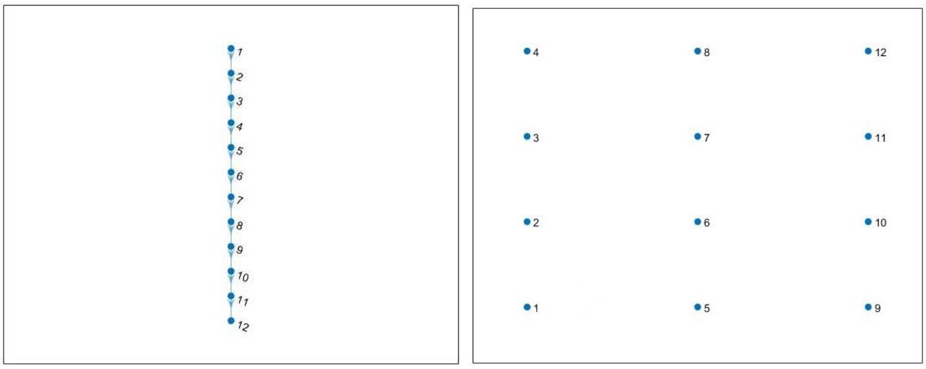
4.2 Task time index $t_{\%}$

The *task time index*, $t_{\%}$, is based on the model input data, i.e. the disparity between the resources task time, which depends on the process adopted to assembly the considered product. Since it is possible to observe an increase in the collaboration between the resources when their task times are comparable, the necessity to study this parameter is reasonable. $t_{\%}$ is evaluated as the ratio between the minimum makespan, evaluated by assigning all tasks to the respective fastest resource, and the maximum one, evaluated by assigning all tasks to the respective slowest resource:

$$t_{\%} = \frac{min(\sum_{j=1}^{J} t_{j1}, ..., \sum_{j=1}^{J} t_{jK})}{max(\sum_{j=1}^{J} t_{j1}, ..., \sum_{j=1}^{J} t_{jK})}, \quad t_{\%} \in (0,1] \tag{18}$$

This formulation takes into account a single parameter for both resources since it is independent of the k index. Similarly to the $p_{\%}$ index, the value of $t_{\%}$ is limited between 0 and one where:

- if $t_{\%}$ is very small, albeit not null, the disparity between the resources task time is high. A null value cannot be reached unless the meaningless scenario where one of the resources have null task times;



(a) $p_{\%} = 0$ (b) $p_{\%} = 1$

Fig. 2: Precedence graphs corresponding to the extreme cases for the parallelism index $p_{\%}$

- if $t_{\%}$ is equal to 1, the resources task times are equal on average, thus this scenario is likely to increase the collaboration between the resources.

Indeed, since the number of tasks is the same for the resources, comparing the sum of the task times or the mean task time is equal. This is shown in Figure 3, where considering a faster operator on average (dotted line in red, 5.42 s for the operator, 6.75 s for the cobot) does imply a smaller sum (65 s for the operator and 81 s for the robot) but not that all the task times are shorter, such as tasks 9 and 10.

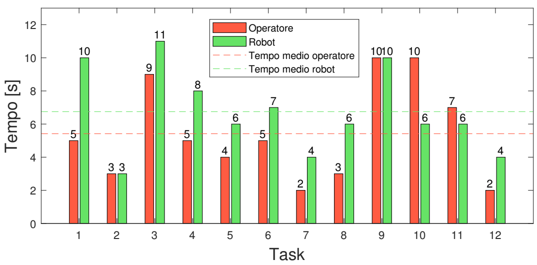


Fig. 3: Example of resources task times for $t_{\%} = 0.8$. Although the mean operator task time is shorter than the robot one, not all task times are shorter.

4.3 Makespan index $m_{\%}$

To evaluate the effectiveness of the scheduling proposed by the model, it is important to compare the obtained ms with a reference parameter. Hence, the makespan index $m_{\%}$ is evaluated as the ratio between the estimated makespan ms and the achievable makespan for null $p_{\%}$:

$$m_{\%} = \frac{ms}{min\{ms\}_{p_{\%}=0}} \tag{19}$$

where $ms_{p_{\%}=0}$ is the achievable makespan for a null parallelism index, i.e. the makespan evaluated for a precedence graph as seen in Figure 2a. The minimization of this value can be solved by assigning each task to the fastest resource, as shown in Figure 4. This formulation

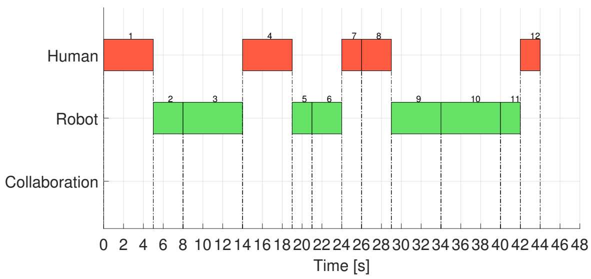


Fig. 4: Task allocation for $p_{\%} = 0$, whose makespan is $ms_{p_{\%}=0}$

allows to compare the makespan obtained with different task allocation methods with a reference one; moreover, it is independent of the task times and J. Lastly, a sub-optimal solution could lead to a value greater than 1, since this index is not limited like the previous ones.

4.4 Collaboration parameter $c_{\%}$

The collaboration parameter $c_{\%}$ represents the interference between the human and the robot during the assembly process. Previous studies presented different formulation for $c_{\%}$ and this work adopts the one presented in [40]:

$$c_{\%} = \frac{T_{coll}}{ms} \quad [0, 1] \tag{20}$$

where T_{coll} is the shared time between the resources, as seen in Figure 5.

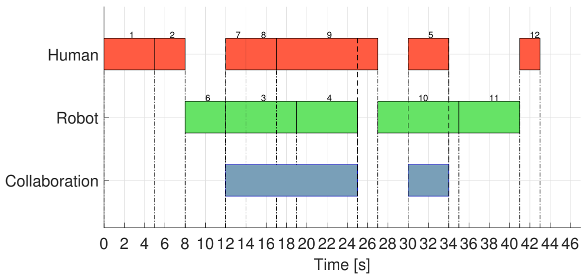


Fig. 5: Example of collaboration: part of the task time is shared between the two agents (in blue).

5 ALB problem models comparison: traditional versus C-ALB

In this section, the proposed model is compared with a variant of the ALB's resolution method studied by Patterson and Albracht [41] by analyzing their resulting task allocation and assembly system performance.

Indeed, the proposed task allocation model is needed since the traditional ALB resolution, which minimizes the cycle time, does not correctly represent a collaborative application, since the modeled layout is a simple assembly line and the constraints do not consider the parallel branches in the precedence graph. Therefore, it is understandable to presume that a task allocation obtained with this model is not optimal to both productivity and collaboration.

To observe the differences between the two models, several precedence graph were considered, to obtain the values presented in Table 2 for the input variables J, $p_{\%}$ and $t_{\%}$. Therefore, the comparison tests were conducted by combining the input data; these values were chosen to point out how the presence of parallel branches and different task times between the resources influence the need for a proper collaborative model.

The task scheduling resulting from the simulations confirm the assumption made before. For example in Table 3, the results obtained with $J = 15$, $p_{\%} = 0.27$, $t_{\%} = 0.4$, are presented: both collaboration time and makespan are smaller when the model used is the one studied for collaborative systems. Moreover, the goodness of the allocation can also be evaluated starting from the makespan index, which is always less than one with the model presented in section 3, while this is not true for the traditional formulation.

J	$p_{\%}$	$t_{\%}$
10	0.27	0.4
15	0.67	0.8

Table 2: Values used for the model comparison.

Table 3: Traditional and collaborative allocation results with $J = 15$ $p_{\%} = 0.27$ e $t_{\%} = 0.4$.

	ALB Patterson-Albracht	**C-ALB** Proposed model	Percentage difference
$m_{\%}$	1,2	0,86	-28%
$c_{\%}$	0,1	0,59	490%
ms $[s]$	77	53	-31%
T_{coll} $[s]$	5	31	520%

The obtained results show that the traditional approach i.e., Patterson-Albracht, that we considered as an example due to its spread, is not suitable to solve the task allocation problem when considering an HRC application. Indeed, the comparison shows a reduction of about 31% when adopting the C-ALB proposed model, and an increase of the shared task time of 520%. This last result shows that adopting traditional approaches that do not consider the collaboration between the resources i.e., parallel task execution, cannot be used for CAS, further proving the necessity of an ALB model designed for CAS. To reach this goal, as previously described, it is fundamental to develop a model that takes into account product characteristics such as $p_{\%}$, which are unique to collaborative systems.

6 Influence of the product and process characteristics on the CAS system

The following paragraph presents the influence of the product and process characteristics i.e., $p_{\%}$ and $t_{\%}$, on the quality of the task allocation for the CAS system, which the results will prove is not negligible. As previously described, $p_{\%}$ depends on the product design, thus it is a product characteristic, whereas $t_{\%}$ represents both the product and process characteristics. As an example we consider a screwing process: the difference in the task times depends on both product characteristics such as screw size, part positioning, and process characteristics, e.g. presence of fixtures, force sensors for the cobots, tools. Moreover, the influence of $t_{\%}$ and especially of $p_{\%}$ is a peculiarity of CAS, which is due to its definition of collaborative resources.

Lastly, the technological constraints in eq. (7) were not considered to avoid adding further parameters in the influence analysis.

6.1 Influence of $p_{\%}$ on $m_{\%}$ and $c_{\%}$

The first tests aim to identify the influence of the precedence graph, represented by $p_{\%}$, on the system performance, i.e. $m_{\%}$ and $c_{\%}$. The tests were carried out with $t_{\%}$ constant and varying J; for each J, 10 values of $p_{\%}$ were identified as seen in Table 4.

Regarding the other input value, $t_{\%}$, the used task times were considered between 2 and 10 s, and with a constant $t_{\%}$ equal to 97%. The obtained results show that the influence of J on the system performance is less significant than the influence of $p_{\%}$, therefore it is not considered in the following graph. The results were fitted with different models, such as hyperbolic, power, and through the coefficient of determination R^2, a 3-rd degree polynomial form was considered the most suitable one, with an R^2 value of at least 0.95. The obtained non-linear equations suggest that the precedence graph could present a modest value of $p_{\%}$ to greatly improve the performance of the system.

Figure 6 represents the effect of $p_{\%}$ on the makespan, represented by the evaluation index $m_{\%}$. As defined, for low values of $p_{\%}$, the makespan is comparable with $ms_{p_{\%}=0}$, thus $m_{\%}$ tends towards 1. Moreover, for a higher value of $p_{\%}$, $m_{\%}$ presents an asymptotic behavior, identified by the red dashed line, with a limit equal to $\frac{1}{K}$, which corresponds to a value of 0.5 for the considered input values. When $m_{\%}$ reaches the asymptotic value, the performance of the system are at their utmost limit and the collaboration parameter $c_{\%}$ is equal to 1.

Figure 7 shows the effect of $p_{\%}$ on the *collaboration parameter* $c_{\%}$. A null collaboration value is correctly expected for a null value of $p_{\%}$, since a similar precedence graph does not allow shared tasks. On the other end, for high value of $p_{\%}$, $c_{\%}$ tends towards the maximum value.

The obtained results prove that a higher value of $p_{\%}$ improves the performance of a CAS since it is possible to execute more tasks in parallel, thus increasing the shared time. Moreover, it proves the need to study the influence of the precedence graph when solving the task allocation problem for a collaborative solution to improve the performance of the system, as seen in Figure 8, where the relation between $c_{\%}$ and $m_{\%}$ is represented.

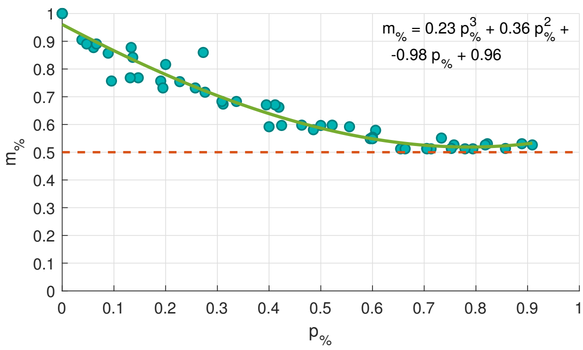


Fig. 6: Effect of the *parallelism index* $p_{\%}$ on the *makespan index* $m_{\%}$.

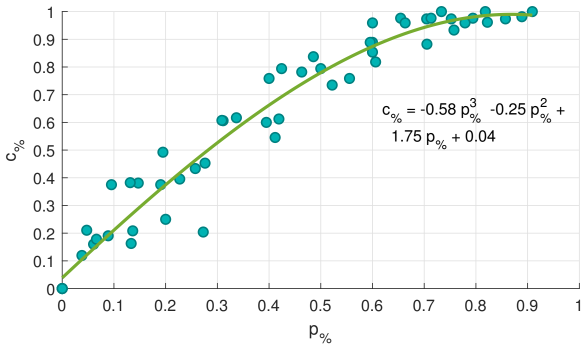


Fig. 7: Effect of the *parallelism index* $p_{\%}$ on the *collaboration parameter* $c_{\%}$

Indeed, the obtained curve proves that an increase in collaboration leads to a lower makespan, therefore proving the necessity of an ALB methodology focused on collaborative application to improve the collaboration, with the improvements shown in the previous section. Lastly, the results hint at the definition of a new product design paradigm i.e., the Design For Collaborative Assembly (DFCA), which aims to design products characterized by branched parallel graphs.

6.2 Influence of $t_{\%}$ on $m_{\%}$ and $c_{\%}$

Differently from $p_{\%}$, it is not possible to distinguish the effect of J and $t_{\%}$, therefore the obtained graph are presented for different values of J. Indeed, for a certain value of $t_{\%}$, a lower value of J means that the difference between the tasks times is less distributed, thus the possibility that the resources are not synchronized is higher, leading to a higher makespan. Therefore, not considering J would lead to an unreliable analysis.

Furthermore, the simulation tests showed the influence of $p_{\%}$ on the effects of $t_{\%}$, even if the overall behavior does not change. Hence, the tests were carried out

J	$p_{\%}$ [%]									
10	0.08	0.13	0.2	0.31	0.4	0.55	0.6	0.73	0.82	0,89
12	0.06	0.13	0.22	0.27	0.42	0.5	0.6	0.75	0.82	0.91
15	0.04	0.1	0.19	0.27	0.42	0.48	0.6	0.71	0.76	0.85
17	0.07	0.14	0.25	0.31	0.41	0.52	0.59	0.65	0.71	0.8
20	0.05	0.13	0.19	0.33	0.39	0.46	0.6	0.66	0.7	0.78

Table 4: Input values used for the test on $p_{\%}$

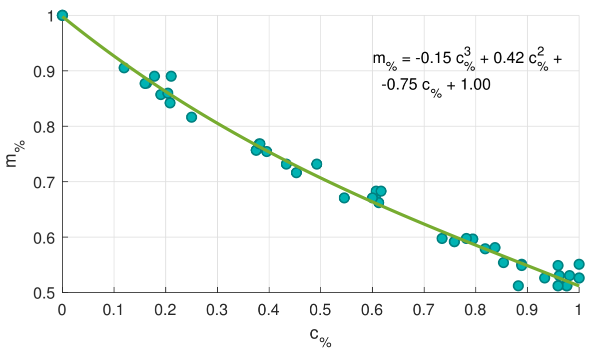


Fig. 8: Relation between $c_{\%}$ and $m_{\%}$ for different values of $p_{\%}$

for different values of $p_{\%}$, represented in Table 5, where for each value of J, 3 values of $p_{\%}$ were considered to represent a low, medium, and a high degree of parallelism. For each J, 21 values for $t_{\%}$ defined between 0.2 and 1 were considered; these are omitted for shortness.

J	$p_{\%}$		
10	0.27	0.488	0.667
12	0.27	0.49	0.67
15	0.27	0.48	0.7

Table 5: Values of $p_{\%}$ used for the tests on $t_{\%}$

Similar to the previous tests, the results were fitted adopting a 3-rd degree polynomial form, since it resulted in a higher value of R^2. Figure 9 represents the influence of $t_{\%}$ for low values of $p_{\%}$ i.e., lower than 0.27. As previously seen with the $p_{\%}$-tests, the non-linear behavior shows that modest changes in $t_{\%}$ greatly improves the system performance; however, in this case for moderate task time difference i.e., $t_{\%}$ greater than 0.6, $m_{\%}$ is constant, thus $t_{\%}$ does not influence the makespan.

The effects of J depends on the adopted value of $p_{\%}$, as seen in Figure 10: for $p_{\%}$ about 0.49, the effect of J is limited with respect to the behaviour shown for $p_{\%}$ about 0.27. This shows that for low values of parallelism a higher number of tasks J improves the system performance, represented by lower values of $m_{\%}$. Indeed, even if $p_{\%}$ is low, a higher J allows to carry out a higher number of tasks, thus reducing the makespan, further confirming the importance of considering parallel tasks when defining the task allocation for collaborative workcells.

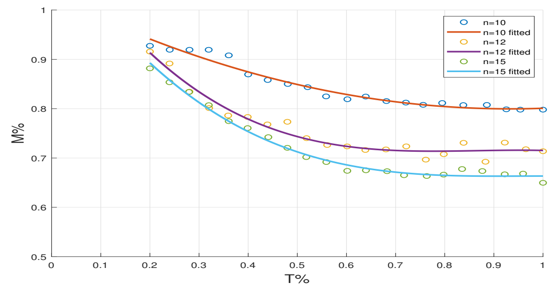


Fig. 9: Effect of the *task time index* $t_{\%}$ on the makespan for low value of $p_{\%}(\approx 0.27)$

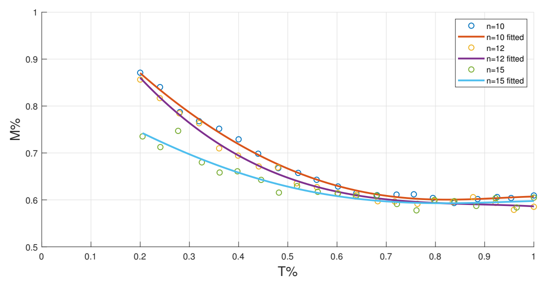


Fig. 10: Effect of the *task time index* $t_{\%}$ on the makespan for medium value of $p_{\%}(\approx 0.49)$

This is further proved by Figure 11, where the influence of J is minimal for a moderate task time difference ($t_{\%}$ greater than 0.6). Moreover, an increased $p_{\%}$ leads to a lower value of $m_{\%}$ in the plateau region, further proving the impact of $p_{\%}$ on the system performance. Lastly, the results did not show a great influence between $c_{\%}$ and $t_{\%}$.

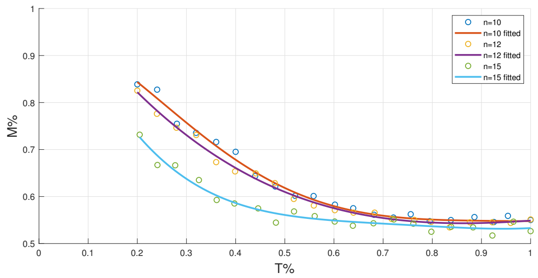


Fig. 11: Effect of the *task time index* $t_{\%}$ on the makespan for medium-high value of $p_{\%}(\approx 0.67)$

6.3 Overall results

The overall behaviour is presented in Figure 12, where the influence of both $p_{\%}$ and $t_{\%}$ on $m_{\%}$ is presented, considering a product with $J = 10$ as an example. The overall behavior is represented by adopting a linear interpolation between the scenarios; although it is not the most faithful fit, it immediately shows the effect of the product and process characteristics on the system performance, which was represented more faithfully in the previous figures.

The obtained results show that it is necessary to take into account the difference of the task times and the precedence graph parallelism to improve the system performance. Moreover, it further proves that, although the graph shows a steeper slope along the direction of $t_{\%}$ than that of $p_{\%}$, which hints to a greater influence of $t_{\%}$, a higher value of $p_{\%}$ allows to achieve better improvements, especially in the case of a limited number of tasks, as represented in the proposed scenario.

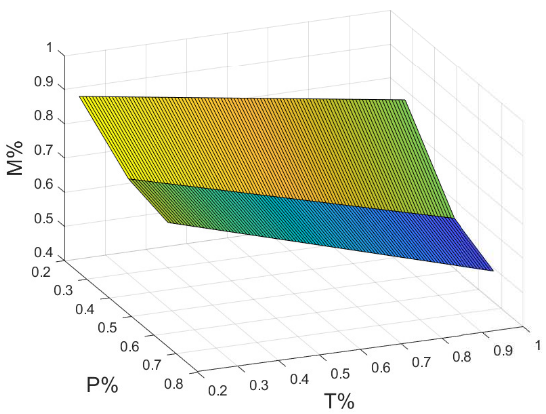


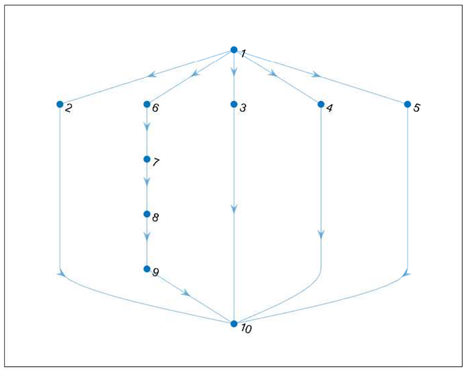
Fig. 12: Influence of the product characteristics $p_{\%}$ and $t_{\%}$ on $m_{\%}$ for $J = 10$

7 Case study

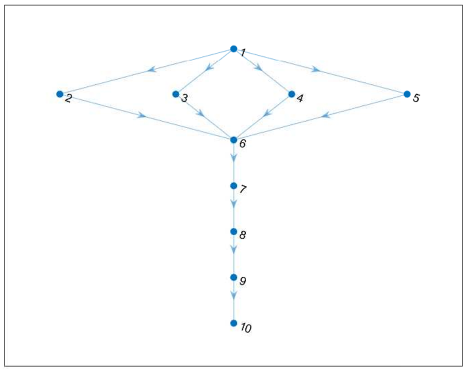
The following section presents an industrial application of the proposed model to demonstrate its practical implication; the considered assembly is a simplified adaptation of a real industrial process. This simplification allowed us to consider two different values of $p_{\%}$ without changing the tasks.

The considered product is defined by a number of tasks J equal to 10 and with the following input characteristics:

- $p_{\%}$ equal to 0.13 (Figure 13a) and 0.48 (Figure 13b)
- $t_{\%}$ equal to 0.13 and 0.91



(a) $p_{\%} = 0.13$



(b) $p_{\%} = 0.48$

Fig. 13: Precedence graph for the considered product for $p_{\%}$ equal to 0.13 and 0.48 respectively.

Regarding $t_{\%}$, in this work, the operator task times were taken from experimental evaluations and thus considered constant, whereas the robot ones were evaluated on the basis of the chosen value of $t_{\%}$ and assigned to the manipulator. This is due since it was difficult to change the operator's speed, differently to the robot. The adopted task times are presented in Table 6.

Task	**Operator task times**	**Robot task times $t_{\%}$ = 0.48**	**Robot task times $t_{\%}$ = 0.60**
1	1	3	3
2	3	6	5
3	3	6	5
4	3	6	5
5	3	6	5
6	7	14	10
7	7	14	10
8	7	14	10
9	7	14	10
10	10	22	20

Table 6: Operator and robot task times for different values of $t_{\%}$

A collaborative workstation has been developed in the Robotics and Automation Laboratory at the Department of Management and Engineering of the University of Padova, as represented in Figure 14. The tests were carried out using a KUKA LBR iiwa 14R820 cobot to perform the tasks assigned offline by the proposed model.

Fig. 14: Experimental tests on the developed workcell.

As a result, Figure 15 presents the task allocation for the case study: the points marked with a red cross are the ones assigned to the human operator, the others are carried out by the cobot.

8 Conclusion

Collaborative assembly systems are increasingly common in the industry since collaborative robots are considered an optimal solution to increase both the flexibility and throughput of the system. However, using traditional ALB models is not optimal since a CAS workstation is composed of multiple resources working in parallel in the same workstation. Hence, this work presents a model for assembly line balancing, called C-ALB, developed for collaborative systems. From the proposed model, the following results could be obtained:

- A proper model for collaborative systems should be considered; indeed, when comparing with traditional ALB models, our C-ALB model achieves a greater throughput and collaboration.
- A difference in the task time between the resources has a minor influence on the system performance unless it is considerable ($t_{\%}$ less than 0.4). Moreover, the number of tasks influences the performance for a lower value of $p_{\%}$.
- Regarding $p_{\%}$, precedence graph with more parallel branches are more suitable for CAS since the proposed model shows that $p_{\%}$ has a great influence on $m_{\%}$.

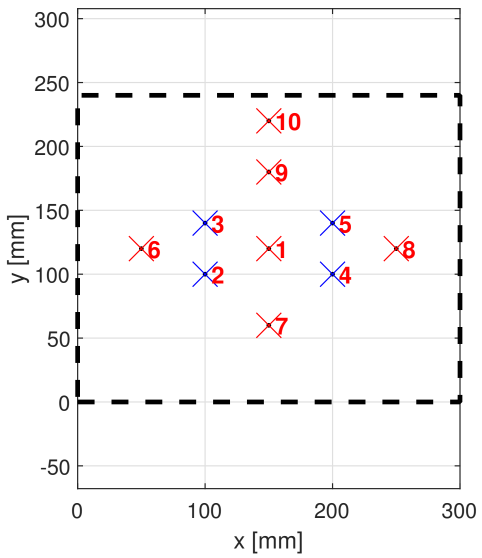


Fig. 15: Task allocation for the case study: the red cross indicates the task assigned to the operator, the blue ones are assigned to the cobot.

As previously observed, this last result shows the importance of the product design when considering a collaborative system; hence a new paradigm called Design for collaborative assembly (DFCA) must be investigated and developed in future works, but this work proves its importance. Further future works will consider CAS composed of multiple resources and workstations in the assembly line.

Contributions The corresponding author M. Faccio has been responsible for planning and coordinating the steps of the research. R. Minto, M. Milanese, and M. Faccio have been responsible for writing and reviewing this paper, together with the quantitative model proposal and the analysis. G.Boschetti has been responsible for revising the paper, including the suitable structure and contents of this paper.

Funding Open Access funding provided by Università degli Studi di Padova.

Compliance with ethical standards

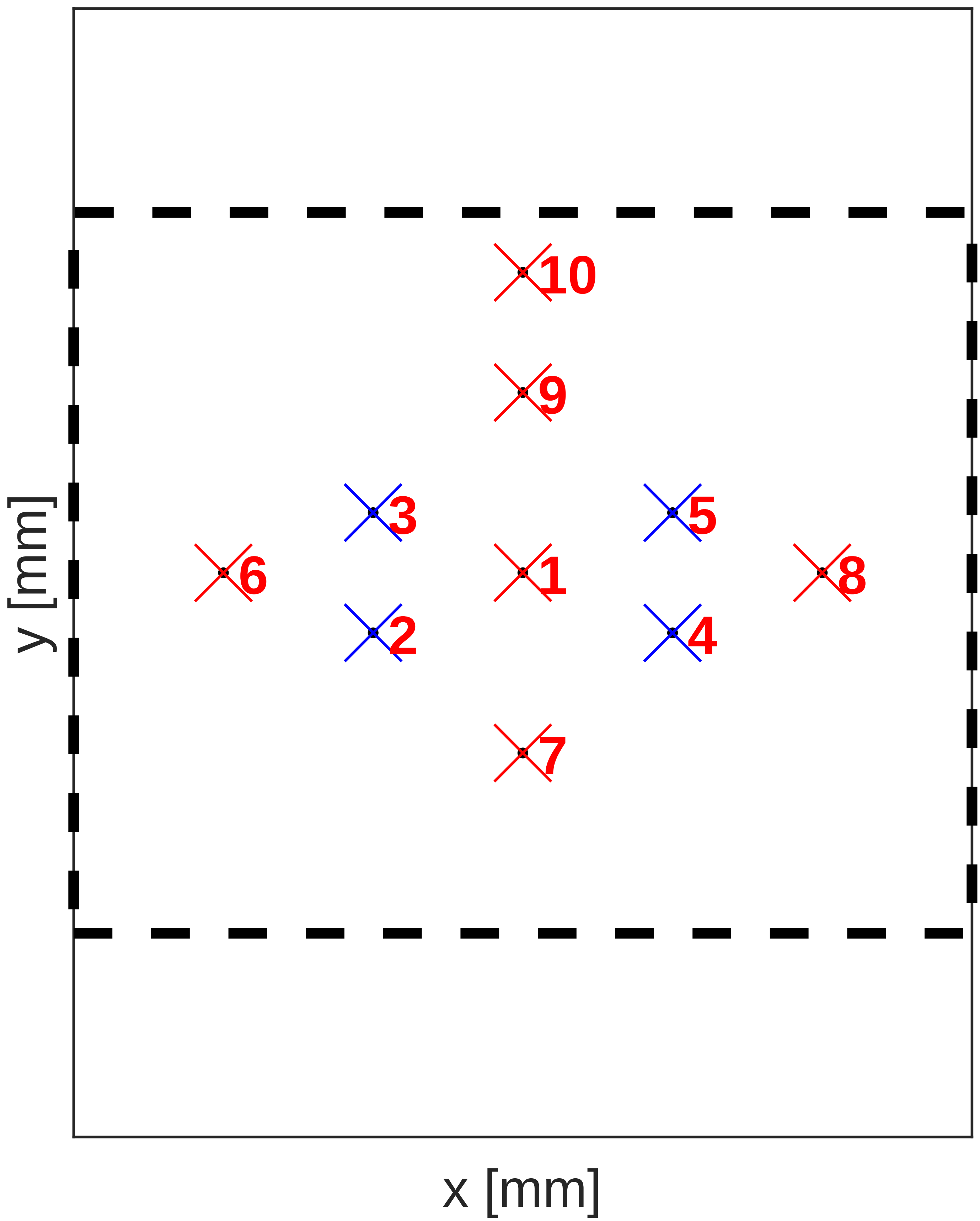
Conflict of interest The authors declare that they have no conflict of interest.
Code availability Not applicable.
Ethical approval This paper is new. Neither the entire paper nor any part of its content has been published or has been accepted elsewhere. It is not being submitted to any other journal as well.
Consent to participate Not applicable

Consent to publish Not applicable

References

1. Nikola Suzić, Cipriano Forza, Alessio Trentin & Zoran Anišić (2018) Implementation guidelines for mass customization: current characteristics and suggestions for improvement, Production Planning & Control, 29:10, 856-871
2. Kim, D. Y. et al. (2020). A modular factory testbed for the rapid reconfiguration of manufacturing systems. Journal of Intelligent Manufacturing, 31(3), 661-680.
3. Azzi A., Battini D., Faccio M., & Persona A. (2012). Sequencing procedure for balancing the workloads variations in case of mixed model assembly system with multiple secondary feeder lines. International Journal of Production Research, 50(21), 6081-6098.
4. Heilala, J., & Voho, P. (2001). Modular reconfigurable flexible final assembly systems. Assembly Automation.
5. Battini, D., Faccio, M., Persona, A., & Sgarbossa, F. (2011). New methodological framework to improve productivity and ergonomics in assembly system design. International Journal of industrial ergonomics, 41(1), 30-42.
6. G. Boothroyd, C. Poli, L. E. Murch, Automatic Assembly. Marcel Dekker, New York (1982)
7. Rosati, G., Faccio, M., Carli, A., & Rossi, A. (2013). Fully flexible assembly systems (F-FAS): a new concept in flexible automation. Assembly Automation.
8. Rosati, G., Faccio, M., Barbazza, L., & Rossi, A. (2015). Hybrid fexible assembly systems (H-FAS): bridging the gap between traditional and fully flexible assembly systems. The International Journal of Advanced Manufacturing Technology, 81(5), 1289-1301.
9. Faccio, M., Bottin, M., Rosati, G. Collaborative and traditional robotic assembly: a comparison model. Int J Adv Manuf Technol 102, 1355–1372 (2019).
10. Takata, S., & Hirano, T. (2011). Human and robot allocation method for hybrid assembly systems. CIRP annals, 60(1), 9-12.
11. Baybars, İ. (1986). A Survey of Exact Algorithms for the Simple Assembly Line Balancing Problem. Management Science, 32(8), 909–932.
12. Scholl, A., & Becker, C. (2006). State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. European Journal of Operational Research, 168(3), 666-693.
13. Becker, C., & Scholl, A. (2006). A survey on problems and methods in generalized assembly line balancing. European journal of operational research, 168(3), 694-715.
14. Pinto, P., Dannenbring, D. G., & Khumawala, B. M. (1975). A branch and bound algorithm for assembly line balancing with paralleling. The International Journal of Production Research, 13(2), 183-196.
15. Cohen, Y., Shoval, S., Faccio, M., & Minto, R. (2021). Deploying cobots in collaborative systems: major considerations and productivity analysis. International Journal of Production Research, 1-17.
16. Tsarouchi, P., Makris, S., & Chryssolouris, G. (2016). Human–robot interaction review and challenges on task planning and programming. International Journal of Computer Integrated Manufacturing, 29(8), 916-931.
17. Johannsmeier, L.,n& Haddadin, S. (2016). A hierarchical human-robot interaction-planning framework for task allocation in collaborative industrial assembly processes. IEEE Robotics and Automation Letters, 2(1), 41-48.
18. Tan, J. T. C., Duan, F., Kato, R., Arai, T., & Hall, E. (2010). Collaboration planning by task analysis in human-robot collaborative manufacturing system. INTECH Open Access Publisher.
19. Salveson M. E. (1955) The assembly line balancing problem. Journal of Industrial Engineering, 6(3): 18-25.
20. Boysen N, Fliedner M, Scholl A (2007) A classification of assembly line balancing problems. Eur J Oper Res 183(2):674–693
21. Michalos, G., Spiliotopoulos, J., Makris, S., & Chryssolouris, G. (2018). A method for planning human robot shared tasks. CIRP journal of manufacturing science and technology, 22, 76-90.
22. Fechter, M., Seeber, C., & Chen, S. (2018). Integrated process planning and resource allocation for collaborative robot workplace design. Procedia CIRP, 72, 39-44.
23. Ranz, F., Hummel, V., & Sihn, W. (2017). Capability-based task allocation in human-robot collaboration. Procedia Manufacturing, 9, 182-189.
24. Chen, F., Sekiyama, K., Cannella, F., & Fukuda, T. (2013). Optimal subtask allocation for human and robot collaboration within hybrid assembly system. IEEE Transactions on Automation Science and Engineering, 11(4), 1065-1075.
25. Bogner, K., Pferschy, U., Unterberger, R., & Zeiner, H. (2018). Optimised scheduling in human–robot collaboration–a use case in the assembly of printed circuit boards. International Journal of Production Research, 56(16), 5522-5540.
26. Nikolakis, N., Kousi, N., Michalos, G., & Makris, S. (2018). Dynamic scheduling of shared human-robot manufacturing operations. Procedia CIRP, 72, 9-14.
27. Pearce, M., Mutlu, B., Shah, J., & Radwin, R. (2018). Optimizing makespan and ergonomics in integrating collaborative robots into manufacturing processes. IEEE transactions on automation science and engineering, 15(4), 1772-1784.
28. Weckenborg, C., Kieckhäfer, K., Müller, C., Grunewald, M., & Spengler, T. S. (2019). Balancing of assembly lines with collaborative robots. Business Research, 1-40.
29. Gualtieri, L., Rauch, E., Vidoni, R., & Matt, D. T. (2019). An evaluation methodology for the conversion of manual assembly systems into human-robot collaborative workcells. Procedia Manufacturing, 38, 358-366.
30. Tsarouchi, P., Spiliotopoulos, J., Michalos, G., Koukas, S., Athanasatos, A., Makris, S., & Chryssolouris, G. (2016). A decision making framework for human robot collaborative workplace generation. Procedia CIRP, 44, 228-232.
31. Weckenborg, C., & Spengler, T. S. (2019). Assembly Line Balancing with Collaborative Robots under consideration of Ergonomics: a cost-oriented approach. IFAC-PapersOnLine, 52(13), 1860-1865.

32. Bruno, G., & Antonelli, D. (2018). Dynamic task classification and assignment for the management of human-robot collaborative teams in workcells. The International Journal of Advanced Manufacturing Technology, 98(9-12), 2415-2427.
33. Malik, A. A., & Bilberg, A. (2019). Complexity-based task allocation in human-robot collaborative assembly. Industrial Robot: the international journal of robotics research and application.
34. Tsarouchi, P., Michalos, G., Makris, S., Athanasatos, T., Dimoulas, K., & Chryssolouris, G. (2017). On a human–robot workplace design and task allocation system. International Journal of Computer Integrated Manufacturing, 30(12), 1272-1279.
35. Heydaryan, S., Suaza Bedolla, J., & Belingardi, G. (2018). Safety design and development of a human-robot collaboration assembly process in the automotive industry. Applied Sciences, 8(3), 344.
36. Antonelli, D., & Bruno, G. (2017). Dynamic task sharing strategy for adaptive human-robot collaborative workcell. DEStech Transactions on Engineering and Technology Research, (icpr).
37. Dianatfar, M., Latokartano, J., & Lanz, M. (2019). Task balancing between human and robot in mid-heavy assembly tasks. Procedia CIRP, 81, 157-161.
38. Yaphiar, S., Nugraha, C., & Ma 'ruf, A. (2019, August). Mixed Model Assembly Line Balancing for Human-Robot Shared Tasks. In Proceedings of the International Manufacturing Engineering Conference & The Asia Pacific Conference on Manufacturing Systems (pp. 245-252). Springer, Singapore.
39. Griva, I., Nash, S. G., Sofer, A. (2009). Linear and nonlinear optimization (Vol. 108). Siam.
40. Faccio, M., Minto, R., Rosati, G. et al. The influence of the product characteristics on human-robot collaboration: a model for the performance of collaborative robotic assembly. Int J Adv Manuf Technol 106, 2317–2331 (2020).
41. Scholl, A. (1999). Balancing and sequencing of assembly lines (No. 10881). Darmstadt Technical University, Department of Business Administration, Economics and Law, Institute for Business Studies (BWL).



10
9
3
5
6
1
8
2
4
7
y [mm]
x [mm]

Figures

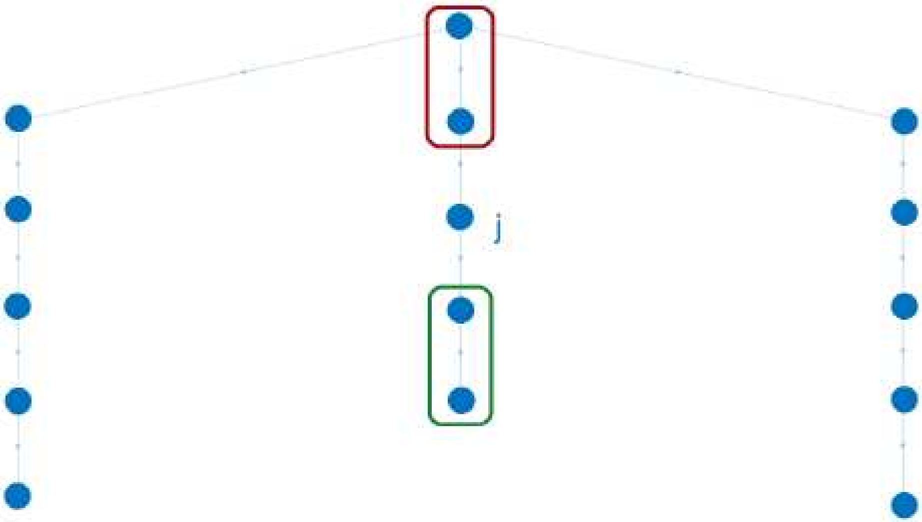


Figure 1

Identification of the Pj predecessors and Sj successors for a task j considering a precedence graph.

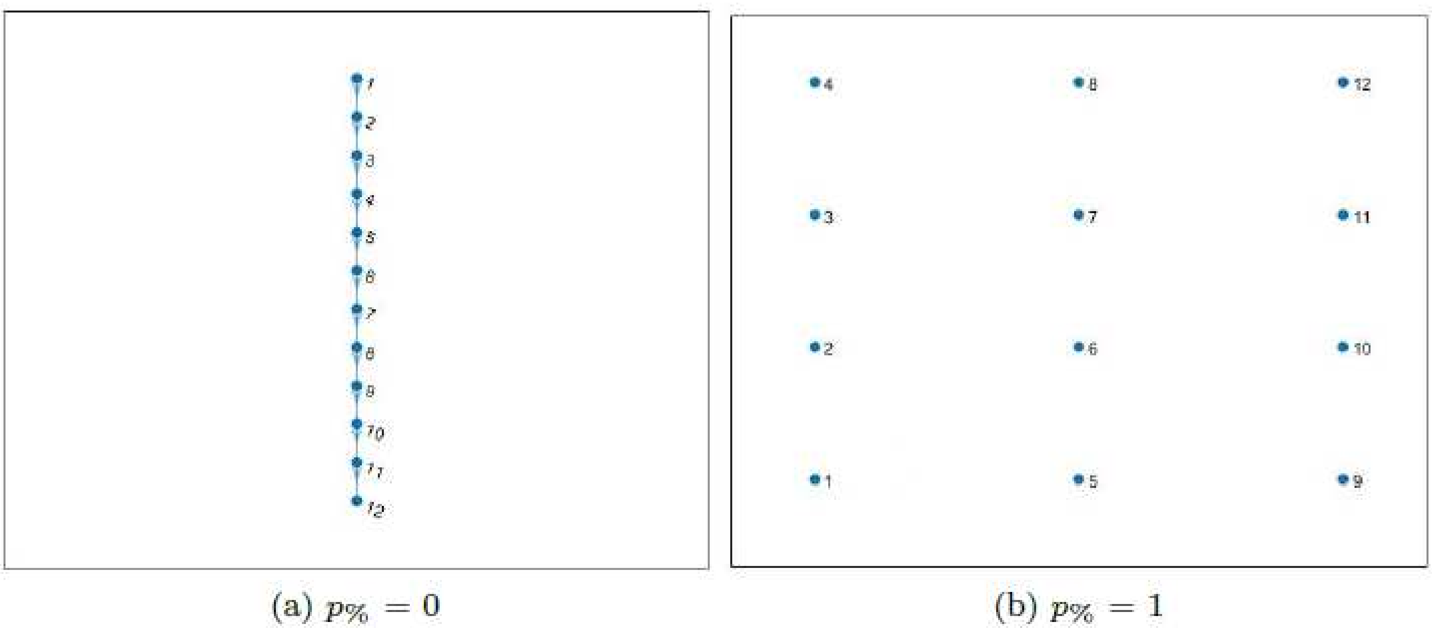


Figure 2

Precedence graphs corresponding to the extreme cases for the parallelism index p%

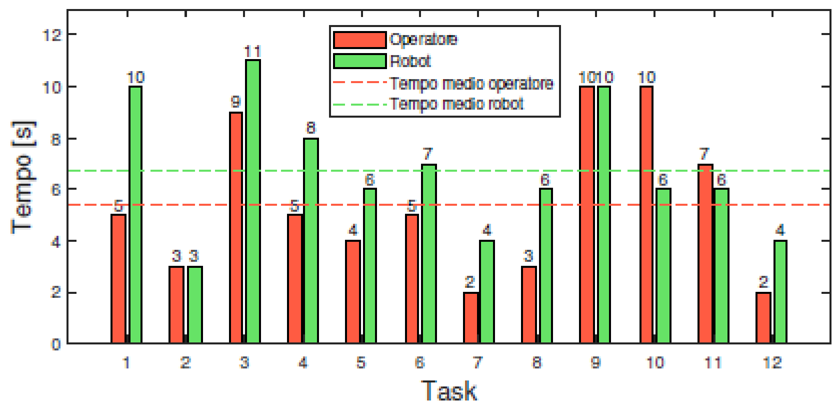


Figure 3

Example of resources task times for t% = 0.8. Although the mean operator task time is shorter than the robot one, not all task times are shorter.

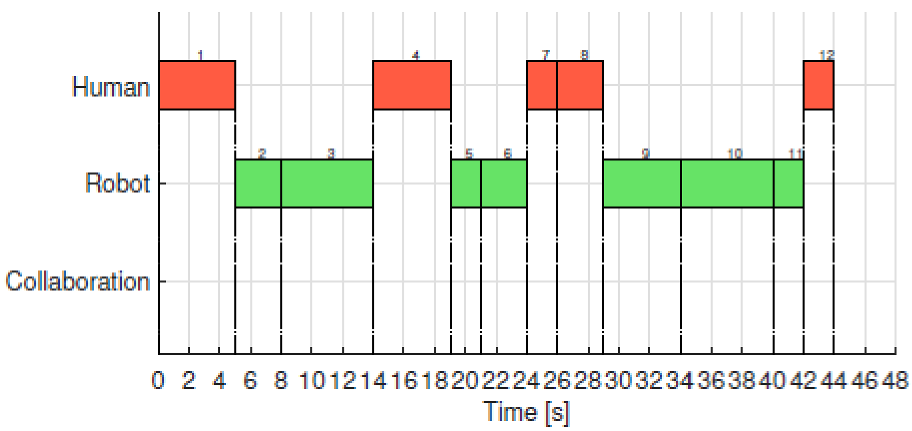


Figure 4

Task allocation for p% = 0, whose makespan is msp%=0

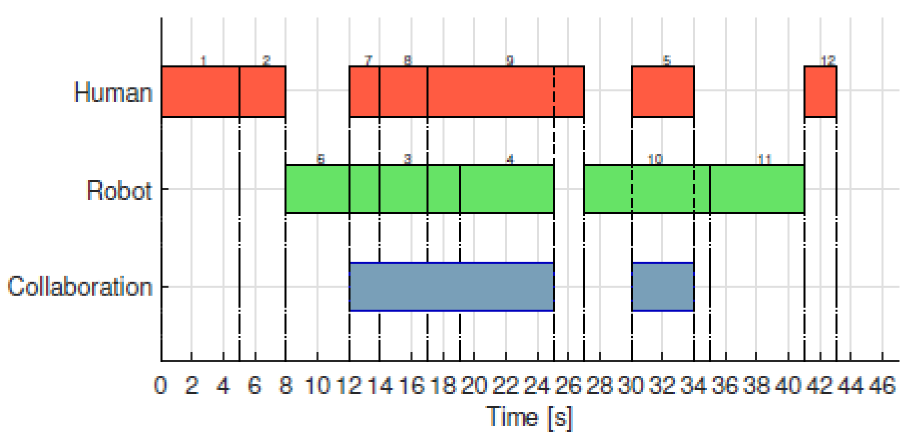


Figure 5

Example of collaboration: part of the task time is shared between the two agents (in blue).

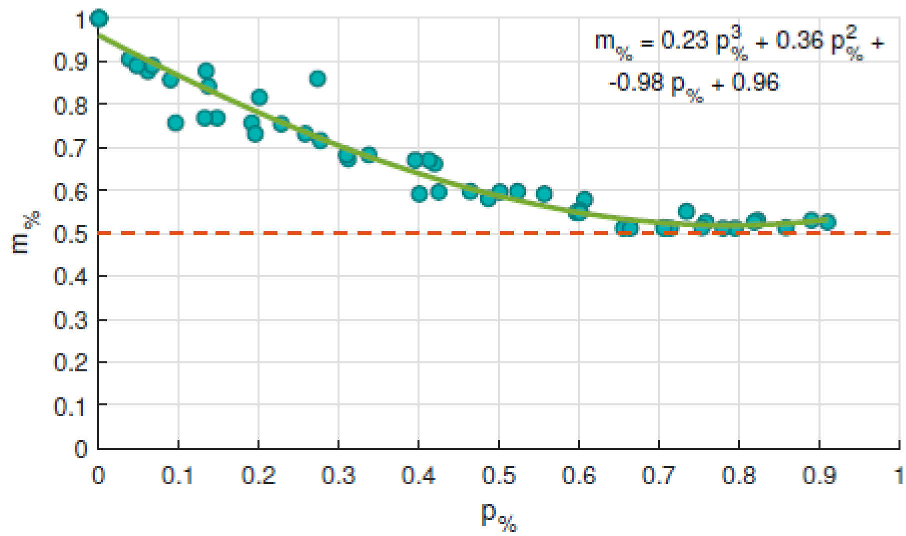


Figure 6

Effect of the parallelism index p% on the makespan index m%.

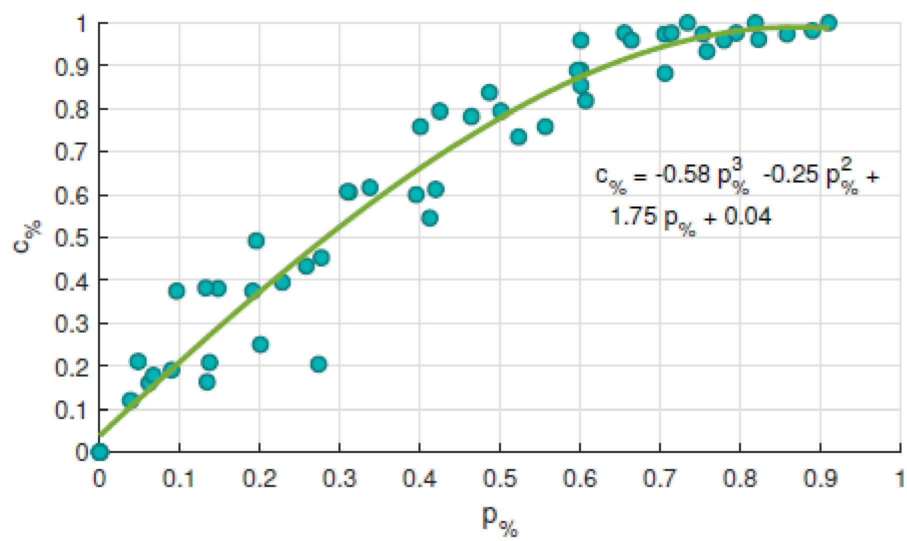


Figure 7

Effect of the parallelism index p% on the collaboration parameter c%

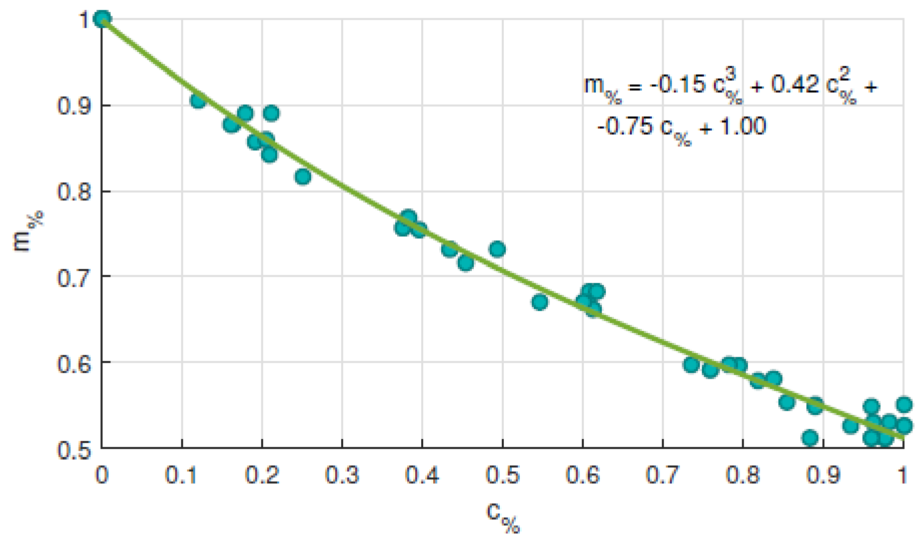


Figure 8

Relation between c% and m% for different values of p%

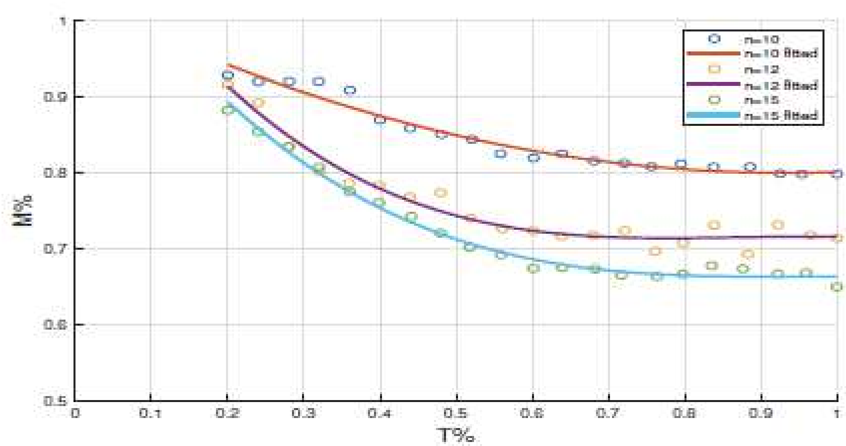


Figure 9

Effect of the task time index t% on the makespan for low value of p%(≈ 0.27)

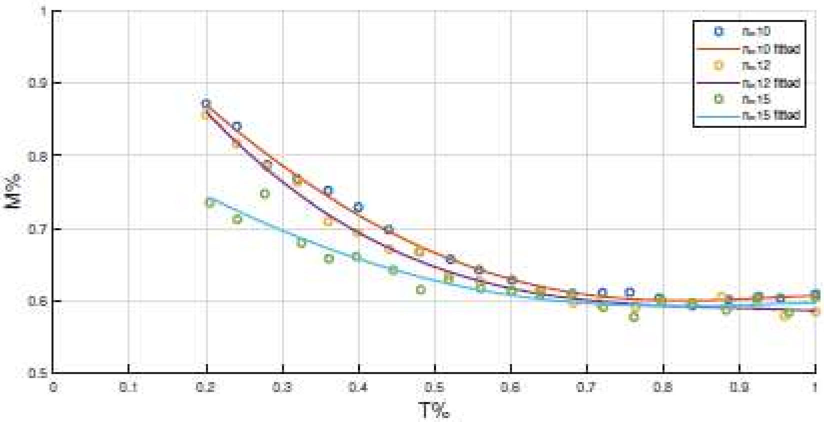


Figure 10

Effect of the task time index t% on the makespan for medium value of p%(≈ 0.49)

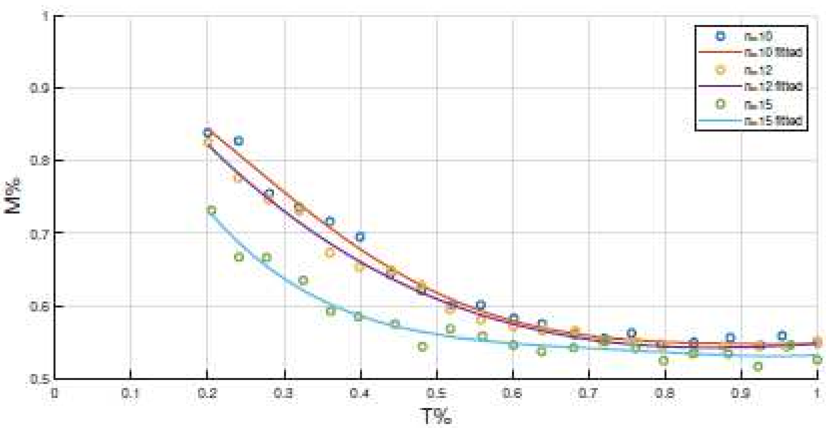


Figure 11

Effect of the task time index t% on the makespan for medium-high value of p%(≈ 0.67)

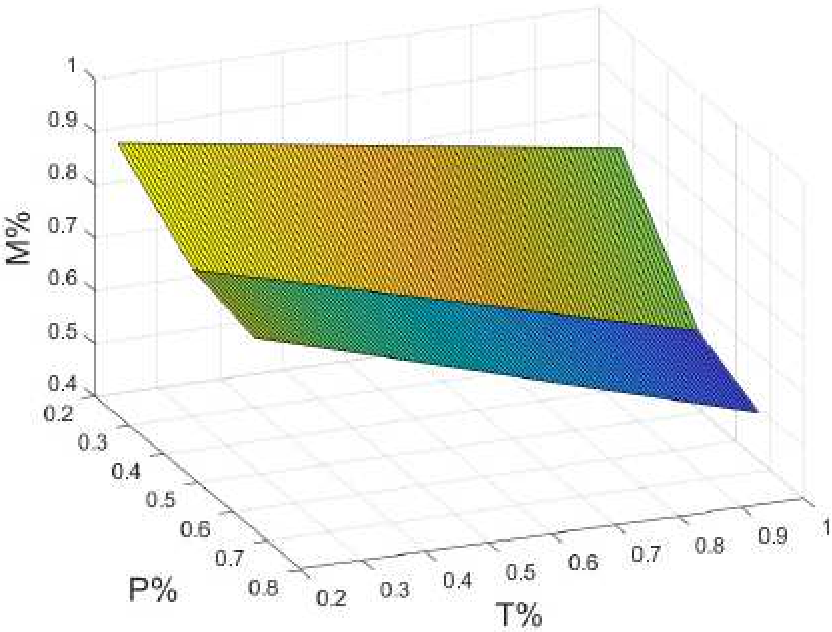
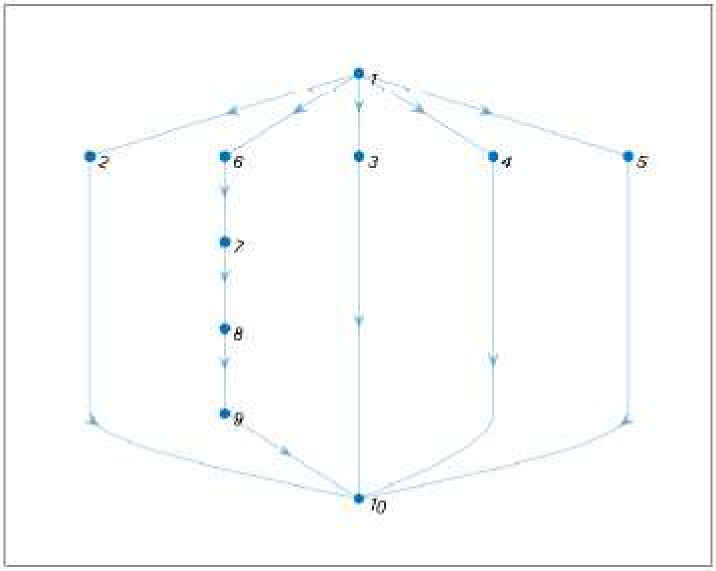
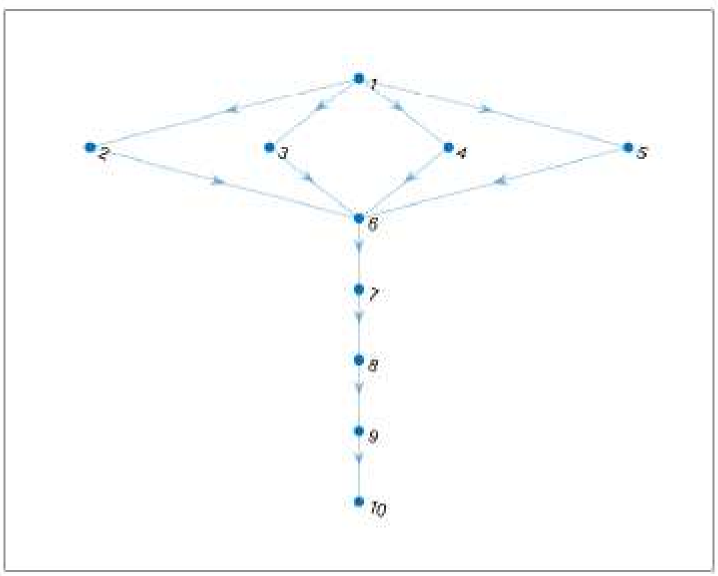


Figure 12

Influence of the product characteristics p% and t% on m% for J = 10



(a) $p\% = 0.13$



(b) $p\% = 0.48$

Figure 13

Precedence graph for the considered product for p% equal to 0.13 and 0.48 respectively.

Figure 14

Experimental tests on the developed workcell.

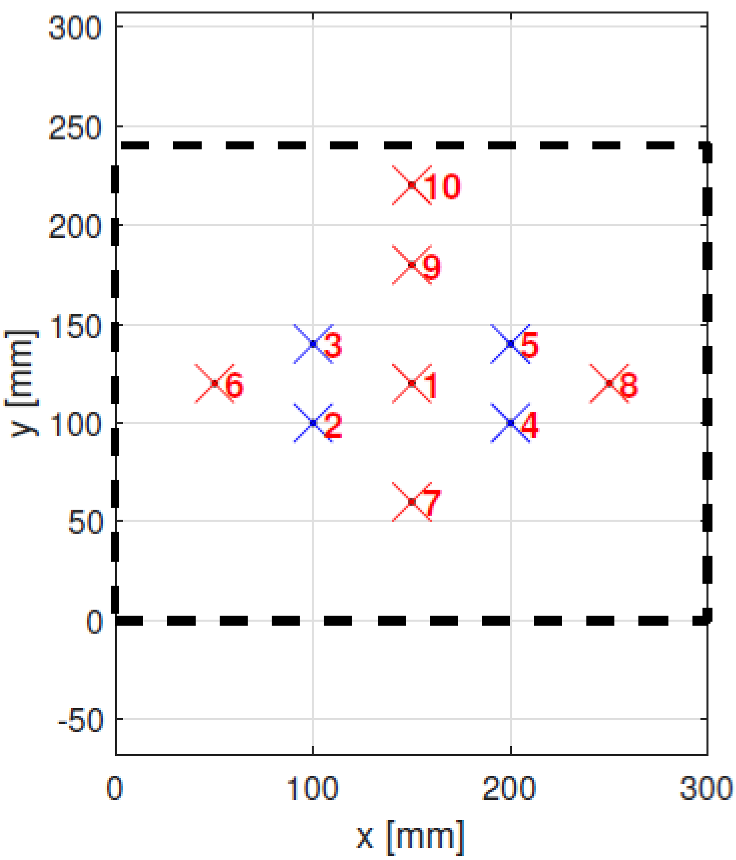


Figure 15

Task allocation for the case study: the red cross indicates the task assigned to the operator, the blue ones are assigned to the cobot.